Statistical pattern recognition for automatic writer identification and verification

Bulacu, Marius Lucian

A modified version of this chapter was published as: Marius Bulacu, Lambert Schomaker, – *"Text-independent writer identification and verification using textural and allographic features,"*, IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), Special Issue - Biometrics: Progress and Directions, IEEE Computer Society, vol. 29, no. 4, pp. 701-717, April 2007

Chapter 5

Feature Fusion for Text-Independent Writer Identification and Verification

If the brain were so simple we could understand it, we would be so simple we couldn't.

Lyall Watson

Abstract

In the previous chapters, we presented the development of new and very effective techniques for automatic writer identification and verification that use probability distribution functions (PDFs) extracted from the handwriting images to characterize writer individuality independently of the textual content of the written samples. This chapter presents a coherent overview of all our features and specifically considers the problem of combining multiple features for text-independent writer identification and verification. Our experiments are also extended to larger datasets containing up to 900 writers.

Our features operate at two levels of analysis: the texture level and the character-shape (allograph) level. For computing the directional texture level features, here we use contours, rather than edges, with definite advantages regarding computation speed and control of feature dimensionality. The contour-based joint directional PDFs encode orientation and curvature information to give an intimate characterization of individual handwriting style. In our analysis at the allograph level, the writer is considered to be characterized by a stochastic pattern generator of ink-trace fragments, or graphemes. The PDF of these simple shapes in a given handwriting sample is characteristic for the writer and is computed using a common shape codebook obtained by grapheme clustering. Combining texture-level and allograph-level features yields very high writer identification and verification performance, with usable rates for datasets containing 10^3 *writers.*

5.1 Introduction

The identification of a person on the basis of scanned images of handwriting is a useful biometric modality with application in forensic and historic document analysis

and constitutes an exemplary study area within the research field of behavioral biometrics. In this chapter, we present an overview of our statistical pattern recognition methods for automatic writer identification and verification using off-line handwriting. We specifically consider the problem of combining multiple features for improving performance on both tasks of writer identification and verification, a topic that was not fully addressed in previous chapters. Here we provide an extensive analysis of feature combinations and report our experimental results obtained on larger datasets containing up to 900 writers.

There are two general characteristics distinguishing our approach: *human intervention is minimized* in the writer identification and verification process and we encode individual handwriting style using *features designed to be independent of the textual content* of the handwritten sample. Writer individuality is encoded using probability distribution functions extracted from handwritten text blocks and, in our methods, the computer is completely agnostic of what has been written in the samples. The development of our writer identification techniques takes place at a time when many biometric modalities undergo a transition from research to real full-scale deployment. Our methods also have practical feasibility and hold the promise of concrete applicability.

Physiological biometrics (e.g. iris, fingerprint, hand geometry, retinal blood vessels, DNA) are strong modalities for person identification due to the reduced variability and high complexity of the biometric templates used. However, these physiological modalities are usually more invasive and require cooperating subjects. On the contrary, *behavioral biometrics* (e.g. voice, gait, keystroke dynamics, signature, handwriting) are less invasive, but the achievable identification accuracy is less impressive due to the large variability of the behavior-derived biometric templates. Writer identification pertains to the category of behavioral biometrics and has applicability in the forensic and historic document analysis fields.

Writer identification is rooted in the older and broader domain of automatic handwriting recognition (Plamondon and Srihari 2000, Vinciarelli 2002). For automatic handwriting recognition, invariant representations are sought which are capable of eliminating variations between different handwritings in order to classify the shapes of characters and words robustly. The problem of writer identification, on the contrary, requires a specific enhancement of these variations, which are characteristic to a writer's hand. Handwriting recognition and writer identification represent therefore two opposing facets of handwriting analysis. It is important, however, to mention also the idea that writer identification could aid the recognition process if information on the writer's general writing habits and idiosyncrasies is available to the handwriting recognition system.

Research in writer identification and verification has received significant interest in recent years due to its forensic applicability (e.g. the case of the anthrax letters). A *writer*

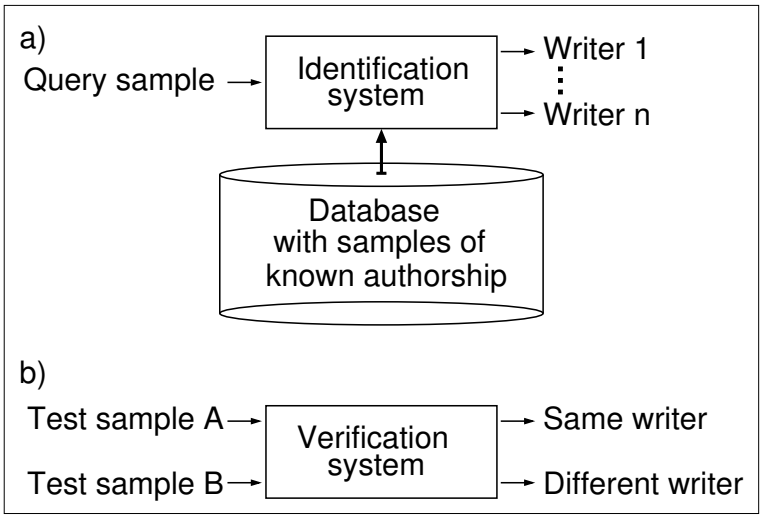

Figure 5.1: *a) A writer identification system retrieves, from a database containing handwritings of known authorship, those samples that are most similar to the query. The hit list is then analyzed in detail by a human expert. b) A writer verification system compares two handwriting samples and takes an automatic decision whether or not the input samples were written by the same person.*

identification system performs a one-to-many search in a large database with handwriting samples of known authorship and returns a likely list of candidates (see Fig. 5.1a). This represents a special case of image retrieval, where the retrieval process is based on features capturing handwriting individuality. The hit list is further scrutinized by the forensic expert who takes the final decision regarding the identity of the author of the questioned sample. Writer identification is therefore possible only if there exist previous samples of handwriting by that person enrolled in the forensic database. *Writer verification* involves a one-to-one comparison with a decision whether or not the two samples are written by the same person (see Fig. 5.1b). The decidability of this problem gives insight into the nature of handwriting individuality. Writer verification has potential applicability in a scenario in which a specific writer must be automatically detected in a stream of handwritten documents. The *target performance* for forensic writer identification systems is a near 100% recall of the correct writer in a hit list of one hundred writers, computed from a database in the order of 10k samples, which is the size of the current European forensic databases. This target performance still remains an ambitious goal. Contrary to other forms of biometric person identification used in forensic labs, automatic writer identification often allows for determining identity in conjunction with the intentional aspects of a crime, such as in the case of threat or ransom letters. This is a fundamental difference from other biometric methods, where the relation between the evidence material and the details of an offense can be quite remote.

Figure 5.2: *A comparison of handwritten characters (allographs) and handwritten words from three different writers. The between-writer variation exceeds the within-writer variability and provides the basis for writer identification and verification.*

Writer identification and verification are only possible to the extent that the variation in handwriting style between different writers exceeds the variations intrinsic to every single writer considered in isolation (see Fig. 5.2). The results reported in this thesis ultimately represent a statistical analysis of the relationship opposing the *between-writer* variability and the *within-writer* variability in feature space. The present study assumes that the handwriting was produced using a natural writing attitude. Forged or disguised handwriting is not addressed in our approach. The forger tries to change the handwriting style usually by changing the slant and / or the chosen letter shapes. Using detailed manual analysis, forensic experts are sometimes able to correctly identify a forged handwritten sample. On the other hand, our proposed algorithms operate on the scanned handwriting faithfully considering all graphical shapes encountered in the image under the premise that they are created by the habitual and natural script style of the writer.

With regard to the theoretical underpinnings of our approach, handwriting can be described as a hierarchical psychomotor process: at a high level, an abstract motor program is recovered from long-term memory; parameters are then specified for this mo-

tor program, such as size, shape, timing; finally, at a peripheral level, commands are generated for the biophysical muscle-joint systems (Maarse 1987). The writer tries to maintain his / her preferred slant and letter shapes over the complete range of motion in the biomechanical systems thumb-fingers and hand-wrist (Maarse 1987) and in a manner that is also independent of changes in the horizontal progression motion (Maarse and Thomassen 1983). Due to neural and neuromechanical propagation delays, a handwriting process based upon a continuous feedback mechanism alone would evolve too slowly (Schomaker 1991). Therefore handwriting is not a feedback process, the brain is continuously planning series of ballistic movements ahead in time in a feedforward manner and a character is assumed to be produced by a "motor program" (Schmidt 1975). Every person uses personalized and characteristic shapes, called *allographs*, when writing a chosen letter of the alphabet (see Fig. 5.2). In this thesis, we propose writer identification methods that aim to capture peripheral and also more central aspects of the writing behavior of an individual. Our methods operate at two levels of analysis: the *texture level* and the *allograph (character-shape) level*. The *texture-level features* are informative for the habitual pen-grip and preferred writing slant, while the *allograph-level features* reveal the character shapes engrained in the motor memory of the writer, as a result of educational, cultural and memetic factors (Schomaker and Bulacu 2004). Furthermore, very effective writer identification and verification is achievable by combining texture-level and allograph-level features, which together offer a fuller description of a person's stable and discriminatory unconscious practices in writing.

This chapter is organized as follows. Section 5.2 describes the datasets used in the experiments reported in this chapter. Sections 5.3 and 5.4 give an overall coherent overview of the algorithms for extracting the texture-level and the allograph-level features respectively. The distances used for feature matching and the feature fusion technique are explained in Section 5.5. Section 5.6 gives the experimental results, followed by a discussion in Section 5.7. Conclusions are then drawn in Section 5.8.

5.2 Experimental datasets

The experiments reported in this chapter were conducted using three datasets: Firemaker, IAM and ImUnipen. The Firemaker and ImUnipen datasets were described previously in the thesis, while IAM is a large dataset newly introduced in this chapter. The IAM dataset (Marti and Bunke 2002) is available on the Internet and was extensively used for off-line handwriting recognition. In addition to the annotation of the textual content, the IAM set contains also writer identity information needed in writer identification studies. For completeness, we provide here brief descriptions of all the datasets

used in the experiments reported in this chapter.

The Firemaker set (Schomaker and Vuurpijl 2000) contains handwriting collected from 250 Dutch subjects, predominantly students, who were required to write 4 different A4 pages. On page 1 they were asked to copy a text of 5 paragraphs using normal handwriting (i.e. predominantly lowercase with some capital letters at the beginning of sentences and names). On page 2 they were asked to copy another text of 2 paragraphs using only uppercase letters. Page 3 contains "forged" text and these samples are not used in the current study. On page 4 the subjects were asked to describe the content of a given cartoon in their own words. These samples consist of mostly lowercase handwriting of varying text content and the amount of written ink varies significantly, from 2 lines up to a full page. The documents were scanned at 300 dpi, 8 bits / pixel, gray-scale. In the writer identification and verification experiments reported in this chapter, we performed searches / matches of page 1 vs. 4 (Firemaker lowercase) and paragraph 1 vs. 2 from page 2 (Firemaker uppercase).

The IAM database (Marti and Bunke 2002) consists of forms with handwritten English text of variable content, scanned at 300 dpi, 8 bits / pixel, gray-scale. Besides the writer identity, the images are accompanied by extensive segmentation and ground-truth information at the text line, sentence and word levels (Zimmermann and Bunke 2002). This dataset includes a variable number of handwritten pages per writer, from 1 page (350 writers) to 59 pages (1 writer). In order to have comparable experimental conditions across all datasets, we modified the IAM set to contain always 2 samples per writer: we kept only the first 2 documents for those writers who contributed more than 2 documents to the original IAM dataset and we have split the document roughly in half for those writers with a unique page in the original set. Our modified IAM set therefore contains lowercase handwriting from 650 persons, 2 samples per writer. The amount of ink is roughly equal in the two samples belonging to one writer, but varies between writers from 3 lines up to a full page.

The ImUnipen set contains handwriting from 215 subjects, 2 samples per writer. The images were derived from the Unipen database (Guyon et al. 1994) of on-line handwriting. The time sequences of coordinates were transformed to simulated 300 dpi images using a Bresenham line generator and an appropriate thickening function. The samples contain lowercase handwriting with varying text content and amount of ink. This set was not directly used in the writer identification and verification tests reported in this chapter. However, a part of this dataset containing 65 writers (130 samples) was used in our allograph-level approach for training the shape codebooks needed for computing the writer-specific grapheme emission probability.

We merged the Firemaker lowercase and IAM datasets to obtain a combined set which we named "Large". The Large dataset therefore contains 900 writers, 2 samples per writer, lowercase handwriting. This combined set is comparable, in terms of number

Table 5.1: *Overview of the experimental datasets, the number of writers contained and some of their properties.*

Dataset	Nwriters	Handwriting	Obs.
Firemaker	250	-lowercase -UPPERCASE	-page 1 and 4 -parag. 1 and 2 of page 2
IAM	650	-lowercase	-original IAM dataset modified to contain 2 samples per writer
ImUnipen	215	-lowercase	-derived from online data, not used in writer identif. and verif. tests, 130 samples by 65 writers used for generating the grapheme codebooks
Large	900	-lowercase	-merger between Firemaker lowercase and IAM datasets

of writers, to the largest dataset used in writer identification and verification until the present (Srihari et al. 2002). It is significant to mention here that our approach to writer identification and verification is text-independent and does not require human effort for labeling. This gave us the noteworthy advantage of being able to easily extend our methods to other datasets and to collect data from multiple sources and different languages in a common framework. Table 5.1 gives an overview of all datasets used in our tests.

5.3 Textural features

Asserting writer identity based on handwriting images requires three main processing phases:

- 1) feature extraction,
- 2) feature matching / feature combination,
- 3) writer identification and verification.

In this and in the following sections of this chapter, we present the feature extraction methods in a general coherent framework. We use probability distribution functions

Table 5.2: *Overview of the considered features, their dimensionalities and the distance functions used in identification and verification. Features are grouped into four different categories: directional PDFs (f1, f2, f3h, f3v), grapheme emission PDF (f4), run-length PDFs (f5h, f5v) and autocorrelation (f6).*

Feature	Explanation	N dims	Dist	Computed from
f1: $p(\phi)$	Contour-direction PDF	12	χ^2	contours
f2: $p(\phi_1, \phi_2)$	Contour-hinge PDF	300	χ^2	contours
	Direction co-occurr.			contours
f3h: $p(\phi_1, \phi_3)$ h	PDFs → horiz. run	144	χ^2	
f3v: $p(\phi_1, \phi_3)$ v	→ vert. run	144	χ^2	
f4: $p(g)$	Grapheme emission PDF	400	χ^2	connected components
	Run-length on white			binary
f5h: $p(rl)$ h	PDFs → horiz. run	60	χ^2	image
f5v: $p(rl)$ v	→ vert. run	60	χ^2	
f6: ACF	Autocorr. horiz.	60	L_2	gray-scale image

(PDFs) extracted from the handwriting images to characterize writer individuality in a text-independent manner. The term "feature" will be used to denote such a complete PDF: not a single value, but an entire vector of probabilities capturing a facet of handwriting uniqueness.

An overview of all the features used in our study is given in Table 5.2. In our analysis, we will consider a number of features that we have designed (*f2*, *f3*, *f4*) and also a number of other features (*f1*, *f5*, *f6*) classically used for writer identification and verification. For the present chapter, we have selected the most discriminative features from a larger number of features tested in a previous paper (normalized entropy, ink-density PDF, wavelets) (Schomaker and Bulacu 2004).

A succession of image processing steps applied on the handwriting image will provide a number of alternate base representations which will then be used for feature computation. The initial gray-scale images containing the scanned samples of handwriting are binarized using Otsu's method (Otsu 1979). The binary images, in which only the ink pixels are "on", undergo connected component detection (labeling) using

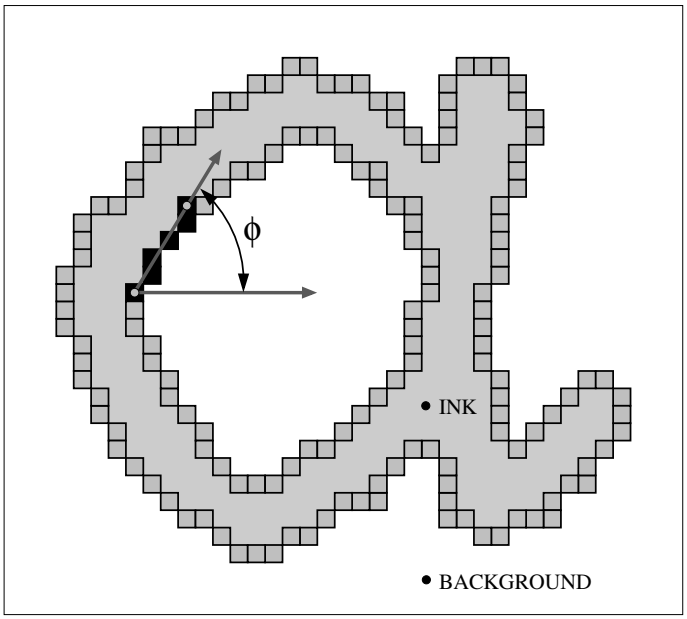

Figure 5.3: *Schematic description for the extraction method of the contour-direction PDF (feature f1). The handwritten letter "a", provided as an example, would be roughly twice as large in reality.*

8-connectivity. Further, for all connected components, the inner and outer contours are extracted using Moore's contour-following algorithm. The contours will contain the sequence of coordinates (x_k, y_k) of all the pixels located exactly on the ink-background boundary. This is a very effective vectorial representation that will allow a fast computation of the directional features. These features were computed using the edge image in the previous chapters of the thesis. Four primary representations of the handwritten document will therefore be used for feature computation: the gray-scale image, the binary image, the connected components and the contours.

The current study implicitly assumes that the foreground / background separation can be realized in a pre-processing phase, yielding a white background with (near-) black ink. This separation will often fail on the smudged and texture-rich fragments sometimes collected in forensic practice, where the ink trace is often hard to identify. However, the complete process of forensic writer identification is never fully automatic and present image processing methods allow for advanced semi-interactive solutions to the foreground / background separation problem.

Our methods work at two levels of analysis: the *texture level* and the *allograph level*. Further in this section, we describe the extraction methods for the texture-level features used in writer identification and verification. In these features, the handwriting is merely seen as a texture described by some probability distributions computed from the image and capturing the distinctive visual appearance of the written samples.

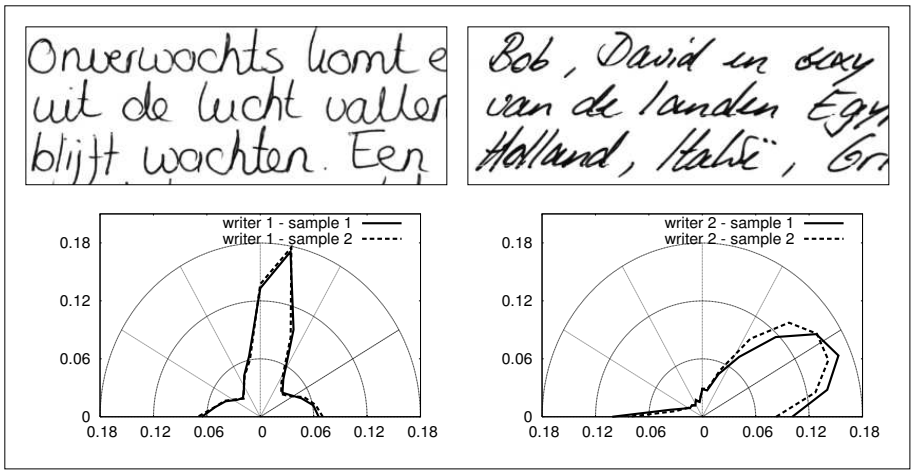

Figure 5.4: *Examples of lowercase handwriting from two different subjects. We superposed the polar diagrams of the direction distribution $p(\phi)$ extracted from the two handwritten samples for each of the two subjects. There is a large overlap between the directional PDFs extracted from samples originating from the same writer, while there is a substantial variation in the directional PDFs for different writers. The examples were chosen for visual clarity.*

5.3.1 Contour-direction PDF (*f1*)

The most prominent visual attribute of handwriting that reveals individual writing style is slant. Handwriting slant is also a very stable personal characteristic (Maarse and Thomassen 1983, Maarse 1987). It has long been known in handwriting research that the distribution of directions in the script provides useful information for writer identification (Maarse et al. 1988), coarse writing-style classification (Crettez 1995) or signature verification (Drouhard et al. 1995). This directional distribution can be computed very fast using the contour representation with the additional advantage that the influence of the ink-trace width is also eliminated.

The contour-direction distribution is extracted by considering the orientation of local contour fragments. The analyzing fragment is determined by two contour pixels taken a certain distance apart (see Fig. 5.3) and the angle that the fragment makes with the horizontal is computed using equation 5.1. As the algorithm runs over the contours, the orientation of the local contour fragments is computed and an angle histogram is built thereby. The angle histogram is then normalized to a probability distribution $p(\phi)$ which gives the probability of finding in the handwriting image a contour fragment oriented at the angle ϕ measured from the horizontal.

$$\phi = arctan(\frac{y_{k+\epsilon} - y_k}{x_{k+\epsilon} - x_k}) \tag{5.1}$$

The parameter ϵ controls the length of the analyzing contour fragment. In our im-

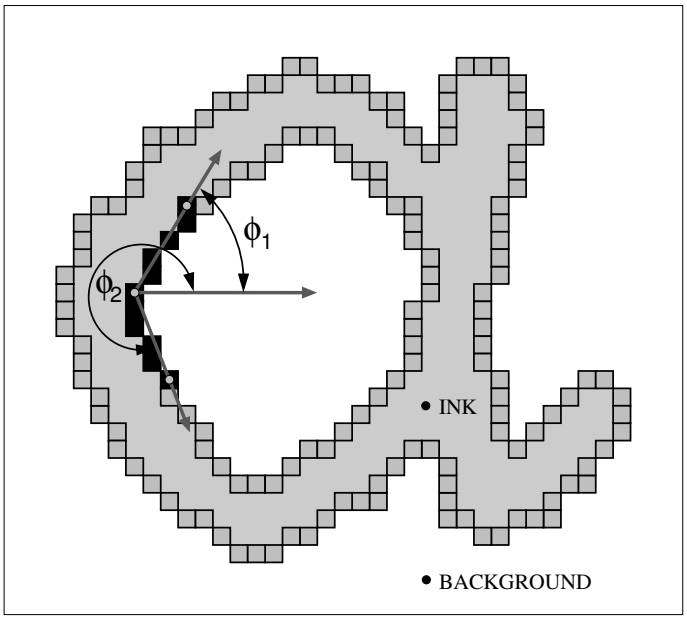

Figure 5.5: *Schematic description for the extraction method of the contour-hinge PDF (feature f2).*

plementation $\epsilon = 5$ and this value was selected such that the length of the contour fragment is comparable to the thickness of the ink trace (6 pixels). The angle ϕ resides in the first two quadrants because, without online information, we do not know which way the writer "traveled" along the probing contour fragment. The number of histogram bins spanning the interval 0° - 180° was set to $n = 12$ through experimentation: 15° / bin gives a sufficiently detailed and, at the same time, sufficiently robust description of handwriting to be used in writer identification and verification. These settings will be used for all the directional features presented in this chapter.

The prevalent direction in $p(\phi)$ (see Fig. 5.4) corresponds, as expected, to the slant of writing. In handwriting recognition, this can be used to deslant the script using a shear transform prior to applying the statistical recognizer. Note that not only the slant (the mode of the angular PDF), but the entire distribution is informative for writer identification. For example, even for the same slant angle, a more round handwriting will have a different directional PDF (more spread) than a more pointed handwriting and it will still be possible to distinguish between them using the distribution $p(\phi)$.

5.3.2 Contour-hinge PDF (*f2*)

The directional distribution $p(\phi)$ represented our starting point in designing more complex features that give a more intimate characterization of the individual handwriting style and ultimately yield significant improvements in writer identification and verifica-

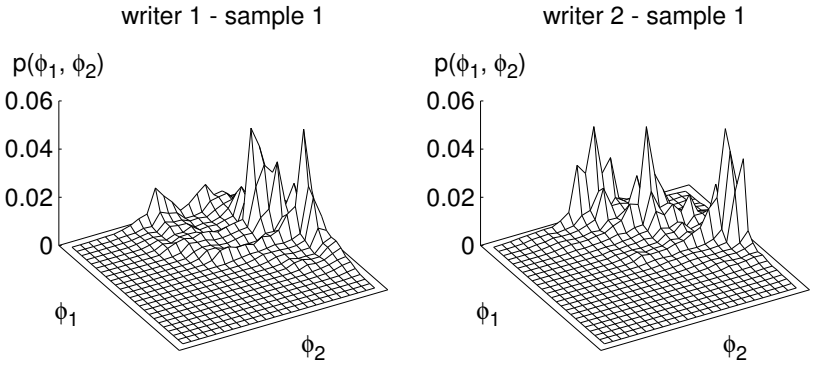

Figure 5.6: *Surface plots of the contour-hinge PDF $p(\phi_1, \phi_2)$ for two writers. One half of the 3D plot (on one side of the main diagonal) is flat because we only consider angle combinations with $\phi_2 \geq \phi_1$.*

tion performance. In order to capture, besides orientation, also the curvature of the ink trace, which is very discriminatory between different writers, we designed the "hinge" feature. The central idea is to consider, not one, but two contour fragments attached at a common end pixel and, subsequently, compute the joint probability distribution of the orientations of the two legs of the obtained "contour-hinge" (see Fig. 5.5). To have an intuitive picture of this feature, imagine having a hinge laid on the surface of the image. Place its junction on top of every contour pixel, then open the hinge and align its legs along the contour. Consider the angles ϕ_1 and ϕ_2 that the legs make with the horizontal and count the found instances in a two dimensional array of bins indexed by ϕ_1 and ϕ_2. The final normalized histogram gives the joint PDF $p(\phi_1, \phi_2)$ quantifying the chance of finding in the image two "hinged" contour fragments oriented at the angles ϕ_1 and ϕ_2 respectively.

In contrast with feature *f1* for which spanning the upper two quadrants (180°) was sufficient, we now have to span all the four quadrants (360°) around the central junction pixel when assessing the angles of the two fragments. The orientation is now quantized in $2n$ directions for every leg of the "contour-hinge". From the total number of combinations of two angles ($4n^2$) we will consider only non-redundant ones ($\phi_2 \geq \phi_1$). The final number of combinations is $C_{2n}^2 + 2n = n(2n + 1)$. For $n = 12$, the contour-hinge feature vector will have 300 dimensions.

The feature $p(\phi_1, \phi_2)$ is a bivariate PDF capturing both the orientation and the curvature of contours. Examples are given in Fig. 5.6. Additionally, the joint probability $p(\phi_1, \phi_2)$ is proportional to the conditional probability $p(\phi_2|\phi_1)$ that can be interpreted as the transition probability from state ϕ_1 to state ϕ_2 in a simple Markov process. Feature *f2* is highly discriminative and gives very satisfying results in writer identification.

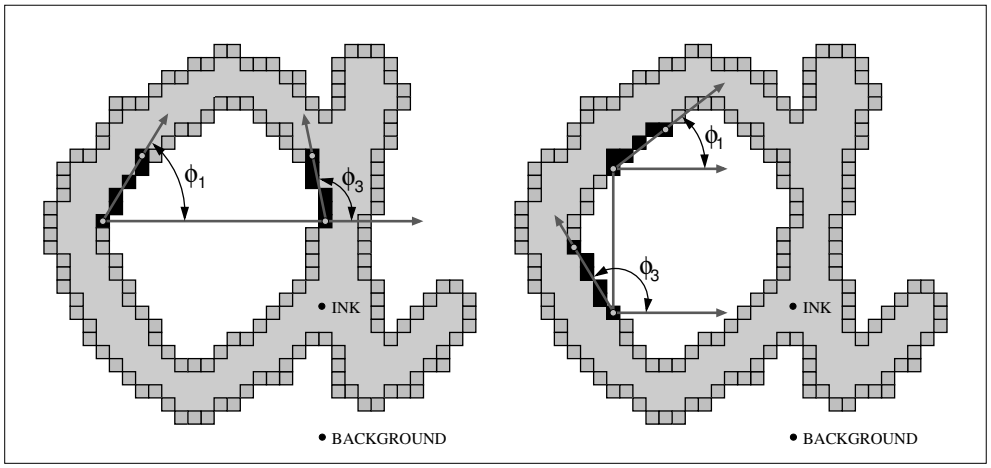

Figure 5.7: *Schematic description for the extraction methods of the direction co-occurrence PDFs (horizontal scan - feature f3h on the left and vertical scan - feature f3v on the right).*

5.3.3 Direction co-occurrence PDFs (*f3h*, *f3v*)

Building upon the same idea of combining oriented contour fragments, we designed another feature: the directional co-occurrence PDF. For this feature, we consider the combination of contour-angles occurring at the ends of run-lengths on the background (see Fig. 5.7). The joint PDF $p(\phi_1, \phi_3)$ of the two contour-angles occurring at the ends of a run-length on white captures longer range correlations between contour directions and gives a measure of the roundness of the written characters. Horizontal runs along the rows of the image generate *f3h* and vertical runs along the columns of the image generate *f3v*. The PDFs *f3h* and *f3v* have n^2 dimensions, namely 144 in our implementation.

These features derive conceptually from the directional distribution *f1* presented above and the run-length distributions *f5h* and *f5v* which will be described further. Examples of $p(\phi_1, \phi_3)h$ for two writers are given in Fig. 5.8.

The features presented thus far (*f1*, *f2* and *f3*) are directional PDFs constructed using oriented contour fragments that act like local phasors and perform, in Fourier terms, a local phase analysis at the scale of the ink-trace width. The local phase correlations are collected in the joint probability distributions that are generic texture descriptors characterizing individual handwriting style independently of the text content of the written samples.

5.3.4 Other texture-level features: run-length PDFs (*f5h*, *f5v*), autocorrelation (*f6*)

Run lengths were first proposed for writer identification in (Arazi 1977, Arazi 1983) and were also used on historical documents in (Dinstein and Shapira 1982). Run lengths

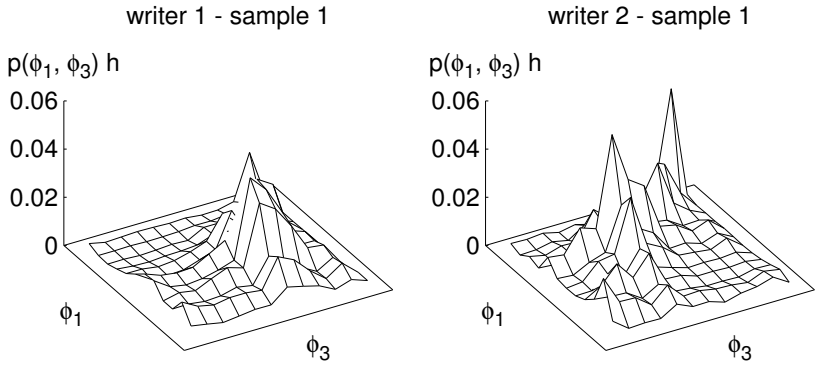

Figure 5.8: *Surface plots of the contour-direction co-occurrence PDF $p(\phi_1, \phi_3)h$ for two writers. Every writer has a different "probability landscape".*

are determined on the binary image taking into consideration either the black pixels corresponding to the ink trace or the white pixels corresponding to the background. The statistical properties of the black runs are significantly influenced by the ink width and therefore by the type of pen used for writing. The white runs capture the regions enclosed inside the letters and also the empty spaces between letters and words. The probability distribution of white lengths (runs on background) will be used in our writer identification and verification tests. There are two basic scanning methods: horizontal along the rows of the image (*f5h*) and vertical along the columns of the image (*f5v*). Similarly to the contour-based directional features presented above, the histogram of run lengths is normalized and interpreted as a probability distribution. Our particular implementation considers only run-lengths of up to 60 pixels to prevent the vertical measurements from going in between successive text lines (the height of a written line in our dataset is about 120 pixels).

To compute the autocorrelation feature (*f6*), every row of the image is shifted onto itself by a given offset and then the normalized dot product between the original row and the shifted copy is calculated. The original gray-scale image is used in the computation and the maximum offset ("delay") corresponds to 60 pixels. For every offset, the autocorrelation coefficients are then averaged across all image rows. The autocorrelation function detects the presence of regularity in writing: regular vertical strokes will overlap in the original row and its horizontally shifted copy for offsets equal to integer multiples of the spatial wavelength of handwriting. This results in a large dot product contribution to the final autocorrelation function. Autocorrelation is the only feature in our analysis that is not a probability distribution function and it will require a different distance measure than the other features, Euclidean (L_2 norm) rather than χ^2.

We note here that the autocorrelation and the power spectrum are Fourier transform

pairs. Therefore, in effect, the autocorrelation function performs a Fourier analysis directly in image space along the pixel rows. The amplitude information is retained and averaged across all image rows, while all phase information is discarded. Directional features (*f1*, *f2* and *f3*) are essentially built on local phase information, while autocorrelation encodes only amplitude information. It will be interesting to consider a performance comparison in the experimental results.

The features presented in this section are generic texture-level descriptors that, when applied to handwriting, capture writer individuality, thus providing the basis for writer identification. Their virtue resides in the local computation on the image and, as such, they are generally applicable and do not impose additional constraints. Using the contour representation for extracting the directional distributions offers definite advantages regarding computation speed and control of feature dimensionality. The PDFs can be estimated even from samples with very reduced amounts of written ink. In our data, many handwritten samples contain as little as three lines of text.

5.4 Allographic features

In this section, we briefly reiterate our allograph-level approach to writer identification and verification. Our method, similar to the approach described in (Bensefia et al. 2005b), is based on assuming that the writer acts as a stochastic generator of ink-blob shapes, or graphemes. The probability distribution of grapheme usage is characteristic of each writer and is computed using a common codebook of shapes obtained by clustering. This approach was first applied to isolated uppercase handwriting (Schomaker and Bulacu 2004) and later it was extended to lowercase cursive handwriting by using a segmentation method (Schomaker et al. 2004).

This writer identification and verification method was fully described in the previous chapter of the thesis and involves three processing stages:

1) Handwriting segmentation: the ink is cut at the minima in the lower contour for which the distance to the upper contour is comparable to the ink-trace width (see Fig. 4.4). The graphemes are then extracted as connected components, followed by size normalization to 30x30 pixel bitmaps, preserving the aspect ratio of the original pattern. This segmentation stage makes our allograph-level method applicable to free-style handwriting, both cursive and isolated.

2) Shape codebook generation: grapheme clustering was applied to a training set containing 41k graphemes extracted from 130 samples (65 writers) from the ImUnipen set. On the new Large dataset, the three clustering algorithms used previously will be compared for a large range of codebook sizes: k-means, Kohonen SOM 1D and 2D (Kohonen 1988, Duda et al. 2001). Fig. 5.9 shows three examples of shape codebooks

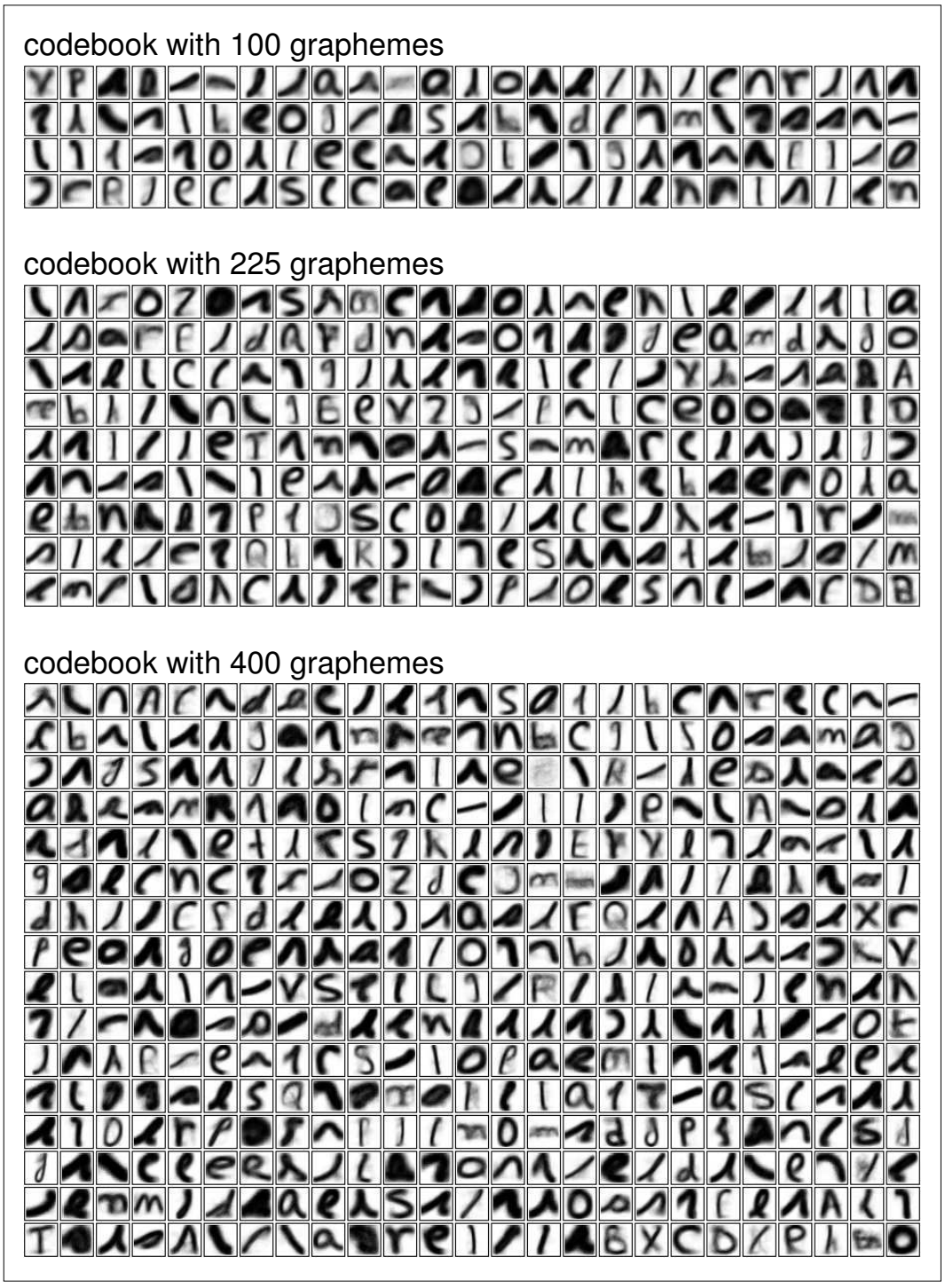

Figure 5.9: *Examples of shape codebooks generated by k-means clustering and containing an increasing number of graphemes (increasing values of the parameter k were used in training).*

generated by k-means clustering for increasing values of k. The codebook graphemes act as prototype shapes representative for the types of shapes to be expected as a result of handwriting segmentation.

3) Grapheme-usage PDF computation: one bin is allocated to every grapheme in the codebook and a shape occurrence histogram is computed for every handwritten sample. For every ink fraglet extracted from a sample after segmentation, the nearest codebook grapheme g is found using Euclidean distance and this occurrence is counted into the corresponding histogram bin. The histogram is normalized to a PDF $p(g)$ that acts as the writer descriptor used for identification and verification.

The perfect segmentation of individual characters in free-style script is still unachievable and this represents a fundamental problem for handwriting recognition. Nevertheless, the ink fraglets generated by our imperfect segmentation procedure can still be effectively used for writer identification. The essential idea is that the ensemble of these simple graphemes still manages to capture the shape details of the allographs emitted by the writer.

The nature of the proposed method does not consist in an exhaustive enumeration of all possible allographic part shapes. Rather, the grapheme codebook spans up a shape space by providing a set of nearest-neighbor attractors for the ink fraglets extracted from a given handwritten sample. The occurrence PDF of these sub-allographic script fragments constitutes a very effective feature for writer identification and verification.

5.5 Feature matching and fusion for writer identification and verification

After the handwritten samples have been mapped onto features capturing writer individuality, an appropriate distance measure between the feature vectors is needed to compute the (dis)similarity, in individual handwriting style, between any two chosen samples. A large number of distance measures were tested in our experiments: Minkowski up to order 5, χ^2, Bhattacharya, Hausdorff. We will report however only on the best performing ones.

For the PDF features (*f1*, *f2*, *f3*, *f4*, *f5*), the χ^2 distance (Press et al. 1992) is used for matching a query sample q and any other sample i from the database:

$$\chi^2_{qi} = \sum_{n=1}^{Ndims} \frac{(p_{qn} - p_{in})^2}{p_{qn} + p_{in}} \tag{5.2}$$

where p are entries in the PDF, n is the bin index and $Ndims$ is the number of bins in the PDF (the dimensionality of the feature). χ^2 is a natural choice as a distance measure for

the PDF features. Euclidean distance is used for the autocorrelation (*f6*).

Writer identification is performed using nearest-neighbor (Cover and Hart 1967) classification in a "leave-one-out" strategy. For a query sample q, the distances to all the other samples $i \neq q$ are computed using a selected feature. Then all the samples i are ordered in a sorted hit list with increasing distance to the query q (Press et al. 1992). Ideally, the first ranked sample should be the pair sample produced by the same writer. If one considers, not only the nearest neighbor (Top 1), but rather a longer list of neighbors starting with the first and up to a chosen rank (e.g. Top 10), the chance of finding the correct hit (the recall) increases with the list size. We point out that, in experiments, we do not make a separation between a training set and a test set, all the data is in one suite. This is actually a more difficult and realistic testing condition, with more distractors: not 1, but 2 per false writer and only one correct hit.

Writer verification, as all biometric verification tasks, can be perfectly placed into the classical Neyman-Pearson framework of statistical decision theory (Neyman and Pearson 1933). For writer verification, the distance ξ between two given handwriting samples is computed using a chosen feature. Distances up to a predefined decision threshold T are deemed sufficiently low for considering that the two samples have been written by the same person. Beyond T, the samples are considered to have been written by different persons. Two types of error are possible: falsely accepting (FA) that two samples are written by the same person when in fact this is not true or falsely rejecting (FR) that two samples are written by the same person when in fact this is the case. The associated error rates are FAR and FRR. In a scenario in which a suspect must be found in a stream of documents, FAR becomes false alarm rate, while FRR becomes miss rate. These error rates can be empirically computed by integrating up-to / from the decision threshold T the probability distribution of distances between samples written by the same person $P_S(\xi)$ and the probability distribution of distances between samples written by different persons $P_D(\xi)$:

$$FAR = \int_0^T P_D(\xi)\, d\xi \tag{5.3}$$

$$FRR = \int_T^\infty P_S(\xi)\, d\xi. \tag{5.4}$$

By varying the threshold T a Receiver Operating Characteristic (ROC) curve is obtained that illustrates the inevitable trade-off between the two error rates. The Equal Error Rate (EER) corresponds to the point on the ROC curve where FAR = FRR and it quantifies in a single number the writer verification performance.

The features considered in the present study are not totally orthogonal, but nevertheless they do offer different points of view on a handwritten sample. It is therefore natural to try to combine them for improving performance (Bulacu and Schomaker 2006),

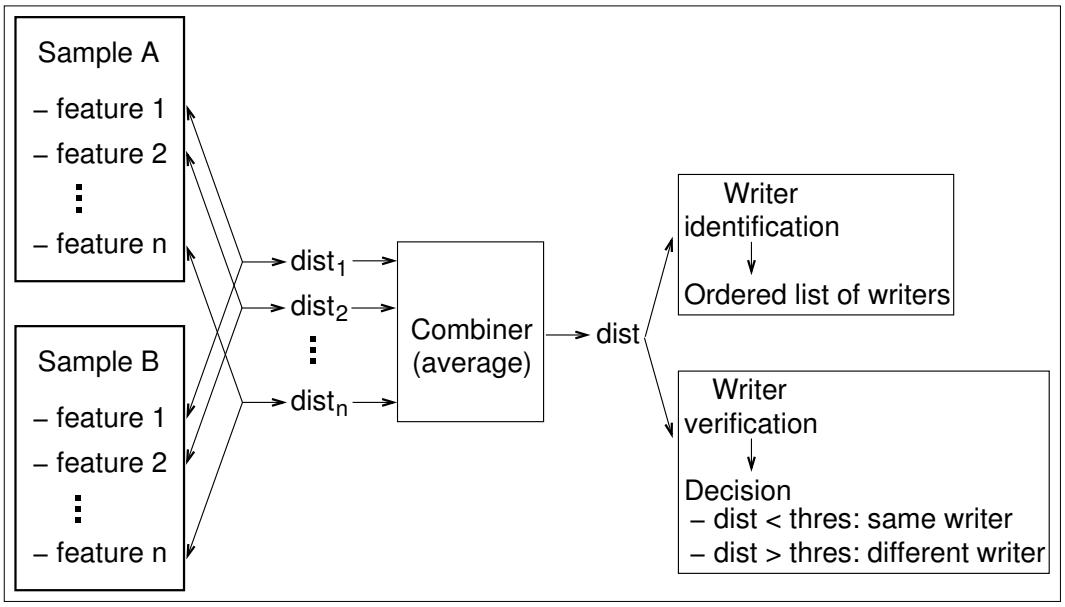

Figure 5.10: *Feature combination scheme: the distances generated by the individual features are averaged (using simple or weighted average) and the result is then used in writer identification and verification.*

this being the main focus of the present chapter of the thesis. In our feature combination scheme, the final unique distance between any two handwritten samples is computed as the average (simple or weighted average) of the distances due to the individual features participating in the combination (see Fig. 5.10).

In feature combinations, Hamming distance performed best:

$$H_{qi} = \sum_{n=1}^{Ndims} |p_{qn} - p_{in}| \tag{5.5}$$

The χ^2 distance, due to the denominator (see eq. 5.2), gives more weight to the low probability regions in the PDFs and maximizes performance for each individual feature. On the other hand, Hamming distance generates comparable distance values for the different PDF features and offers a common ground with slight advantages in feature combinations.

The Bayesian framework underlying the feature combination scheme proposed here entails two fundamental assumptions: features are independent and the probability of two samples having been written by the same person assumes an exponential distribution with respect to the distance between the two samples as generated by a chosen feature $P_S(\xi) \propto e^{-\xi/\sigma}$. The decay constants σ control the weights that different features take on in the combination. While this basic probabilistic model will almost certainly be violated in reality, experimental results show that significant performance improve-

ments are nevertheless achievable by using the proposed feature combination method.

In a more general perspective, feature fusion for writer identification and verification pertains to the broader theme of classifier combination (Kittler et al. 1998) or multi-modal biometrics (Maltoni et al. 2003, Roli et al. 2002). Information can be combined at three levels in the biometric identification or verification process: *sensor fusion*, *similarity-score fusion* and *decision-level fusion* (Daugman 2000). Combining similarity scores (”soft” fusion) seems to be the method of choice in multi-modal biometrics. This is also confirmed in our experiments: we obtained the best feature fusion results by combining the distances (or similarity scores) generated by the individual features.

5.6 Results

In this section of the chapter, we present our experimental results. The performance measures used are the Top-1 and Top-10 identification rates and the Equal-Error-Rate (EER) for verification. As explained in section 5.2 of this chapter, four datasets are considered in the experimental evaluation (see Table 5.1): Firemaker uppercase (250 writers), Firemaker lowercase (250 writers), IAM (650 writers) and Large (900 writers obtained by merging Firemaker lowercase and IAM datasets). All datasets contain 2 samples per writer and writer identification searches are performed in a ”leave-one-out” manner. The shape codebook necessary for computing the grapheme occurrence probability (feature *f4*) was built using part of the ImUnipen dataset (65 writers, 2 samples / writer, 41k bitmap patterns). This ensures a complete separation, at the level of the writers, between the training and the testing data. For the results reported in this section, we used a grapheme codebook generated by k-means clustering and containing 400 prototype shapes.

We are interested in a comparative performance analysis of the different features across the four test datasets. We are also interested in the improvements in performance obtained by combining multiple features. First we shall consider the individual features and then their combinations.

5.6.1 Performances of individual features

Table 5.3 gives the writer identification and verification performance of the individual features considered in this study. While there are important differences in performance among the different features, it can be noticed that, for a chosen feature, performance is consistent across the four experimental datasets. The best performer is the contour-hinge PDF (*f2*) followed by the grapheme-emission PDF (*f4*).

The results obtained on Firemaker uppercase are comparable to those obtained on

Firemaker lowercase. Although the amount of ink contained in the samples varies between the two datasets, this result is nevertheless interesting because, in our data, the uppercase samples generally contain less handwriting than the lowercase ones. Similar results were reported in Chapter 3 in experiments where the amount of ink in the samples was controlled (Bulacu and Schomaker 2003). These findings contradict the idea, one might intuitively expect, that it is always easier to identify the author of lowercase rather than uppercase handwriting. Naturally, the features used are sensitive to major style variations and, in mixed searches (e.g. lowercase query sample / uppercase dataset), performance is very low.

The writer identification performances obtained on Firemaker lowercase and IAM are very similar, albeit the large difference in the number of writers contained in the two datasets. This is probably due to differences in the writer distributions underlying the two datasets. The Firemaker dataset was collected from a rather uniform population in terms of age and education, predominantly Dutch students, and, as a consequence, there is less variation in writing styles compared to the IAM dataset. Under these conditions, when these two datasets are combined, only a slight decrease in writer identification performance on the Large dataset is noticed. The dependence of the writer identification rate on number of writers contained in the dataset is discussed in the following section of this chapter. For the size of the datasets used here, the writer identification percentages are subject to a 3-4% confidence interval at a 95% confidence level.

Table 5.3: *Writer identification and verification performance of individual features. The χ^2 distance was used in matching. The features are explained in Table 5.2.*

	Dataset	Firemaker - UPPERCASE 250 writers			Firemaker - lowercase 250 writers			IAM - lowercase 650 writers			Large - lowercase 900 writers		
	Feature	Top 1	Top 10	EER	Top 1	Top 10	EER	Top 1	Top 10	EER	Top 1	Top 10	EER
f1	$p(\phi)$	43	81	7.6	48	79	7.7	46	76	7.1	43	72	7.1
f2	$p(\phi_1, \phi_2)$	84	96	4.1	81	92	4.8	81	92	5.0	80	91	4.8
f3h	$p(\phi_1, \phi_3)$ h.	51	84	8.3	68	86	6.4	68	87	5.5	65	84	5.9
f3v	$p(\phi_1, \phi_3)$ v.	37	72	16.0	66	89	7.6	65	84	9.6	59	82	9.1
f4	$p(g)$	65	92	8.0	75	92	5.7	80	94	5.6	76	92	5.8
f5h	$p(rl)$ h.	8	32	17.2	18	50	14.4	10	32	17.0	8	29	16.6
f5v	$p(rl)$ v.	9	39	14.8	16	44	16.3	8	31	15.5	10	34	12.1
f6	ACF	16	47	15.6	16	48	15.3	13	38	16.1	12	35	14.7

From the point of view of Fourier analysis, it is important to observe that the contour-direction feature *f1*, encoding local phase information, performs much better than the autocorrelation feature *f6*, encoding amplitude information. In computer vision, it is commonly acknowledged that phase information is predominantly used for identification, while amplitude information is generally used for recognition mainly due to the shift-invariance of the power spectrum. Phase demodulation and phase-based representations are pervasive in biometric identification (Daugman 1993, Jain et al. 1997).

Further more, the contour-angle combination features *f2*, *f3h* and *f3v*, based on local phase correlations, deliver significant improvements in performance over the basic directional PDF *f1*. This confirms the general principle that joint probability distributions do capture more information from the input signal. And, despite their higher dimensionalities, reliable probability estimates can be obtained for the proposed joint PDFs when a few handwritten text lines are available (usually more than three in our datasets). An analysis of writer identification performance vs. amount of ink contained in the samples is given in Chapter 2 (Bulacu et al. 2003).

The run length PDFs, despite having the worst performance among the echelon of features selected in this study, in fact do perform better than a number of other known writer identification features, e.g. entropy, wavelets (see (Schomaker and Bulacu 2004) for a wide analysis).

In brief, our results show that the contour-based angle-combination PDFs (*f2*, *f3h*, *f3v*) and the grapheme-emission PDF (*f4*) outperform the other features over the four test datasets. They constitute the gist of our text-independent approach to writer identification and verification.

5.6.2 Performances of feature combinations

The features considered in this thesis capture different aspects of handwriting individuality and operate at different levels of analysis and also at different scales. While our features are not completely orthogonal, combining multiple features proves, nevertheless, to be beneficial. As stated previously, feature fusion is performed by distance averaging. Assigning distinct weights to the different features participating in the combination yields only very small performance improvements as will be shown further. This has lead us to prefer simplicity and robustness here and report the feature combination results obtained by plain distance averaging.

The features studied here can be grouped into four broad categories (see Table 5.2): contour-based directional PDFs (*f1*, *f2*, *f3h*, *f3v*), grapheme emission PDF (*f4*), run-length PDFs (*f5h*, *f5v*) and autocorrelation (*f6*). We will analyze combinations of features within and between these broad feature groups.

Table 5.4: *Writer identification and verification performance of feature combinations. The $Hamming$ distance was used in matching. Combining features from different feature groups yields improvements in performance over the best individual feature participating in the combination. There is one exception marked with parentheses: Top-1 identification rate for f2 & f4 on Firemaker uppercase dataset.*

Dataset	Firemaker - UPPERCASE 250 writers			Firemaker - lowercase 250 writers			IAM - lowercase 650 writers			Large - lowercase 900 writers		
Combination	Top 1	Top 10	EER	Top 1	Top 10	EER	Top 1	Top 10	EER	Top 1	Top 10	EER
f3: *f3h* & *f3v*	63	89	7.6	75	92	4.8	77	91	5.3	73	89	5.0
f5: *f5h* & *f5v*	29	70	8.1	42	75	9.6	31	60	9.0	33	63	7.5
f1 & *f4*	69	95	5.6	79	93	4.1	84	95	3.3	81	94	3.3
f1 & *f5*	66	93	4.1	70	91	4.6	68	91	4.0	67	90	3.6
f2 & *f4*	(77)	97	4.5	83	94	3.2	88	97	2.8	86	95	2.9
f3 & *f4*	75	95	5.2	82	94	3.8	86	96	3.9	84	95	3.9
f3 & *f5*	79	95	4.0	80	94	3.4	82	94	3.9	80	94	3.7
f4 & *f5*	76	97	4.7	79	94	3.7	85	96	3.1	83	95	3.2
f1 & *f4* & *f5*	82	98	4.0	82	95	3.2	87	96	2.8	85	96	2.8
f2 & *f4* & *f5*	85	98	3.7	83	95	3.2	89	97	2.8	87	96	2.6
f3 & *f4* & *f5*	86	97	3.6	83	95	3.2	89	96	3.3	87	96	3.3

First, we consider the natural combinations *f3h* with *f3v* and *f5h* with *f5v* (first two rows of Table 5.4). Features *f3* and *f5* are therefore obtained by combining the two orthogonal directions of scanning the input image. Compared to their single horizontal or vertical counterparts, the fused features perform markedly better and they will be used, as such, in future combinations.

It is important to note that further combining directional features (*f1* & *f2*, *f1* & *f3*, *f2* & *f3* or *f1* & *f2* & *f3*) did not produce extra improvements over the performance of the best feature involved in the combination. Rather, the experimental results show that improvements are obtained by combining features from different feature groups. In the results given in Table 5.4, the combined performance exceeds the performances of all individual features involved in the combination, with only one exception marked with parenthesis. As can be noticed, the performance of feature combinations is generally consistent over the four experimental datasets.

The best performing feature combinations fuse directional, grapheme and run-length information yielding, on the Large dataset, writer identification rates of 85-87% Top-1 and 96% Top-10 with an EER around 3% for verification.

In Fig. 5.11a, we show the results obtained by considering a weighted combination between features *f2* and *f4*: $d = (1 - \lambda)d_2 + \lambda d_4$, where λ is the mixing coefficient. Similarly, in Fig. 5.11b, we consider the combination *f3* and *f4*: $d = (1 - \lambda)d_3 + \lambda d_4$. Only marginal improvements are attainable over the performance corresponding to simple distance averaging at $\lambda = 0.5$. These results are, in fact, representative for extensive weight optimization tests carried on different feature combinations and generating, in the end, very small overall additional performance improvements.

Such a direct feature combination by simple distance averaging is possible in our case because the fused features are PDFs (that sum up to 1) and, for a chosen pair of samples, the Hamming distances produced by the different features lie roughly within the same range. The only exception is autocorrelation feature *f6* which requires weighting with respect to the other features. This has lead, however, only to minor additional improvements in performance, only about 1% increase in Top-1 identification rate.

We mention that we replaced the linear distance combiner with an SVM (Joachims 1999, Burges 1998, Cristianini and Shawe-Taylor 2000) trained for writer verification. The output of the SVM, i.e. the distance to the separating hyperplane in the space induced by the kernel function, was used for writer identification (ordering the samples with increasing distance) and writer verification (decision same / different writer). The linear kernel outperformed the other general-purpose kernels (polynomial, radial basis, sigmoid). However, the experimental results were rather dismal, not justifying, in our view, the increase in system complexity and computation time.

We also experimented with Borda rank combination schemes in Chapter 3 with only marginal performance improvements (Bulacu and Schomaker 2003).

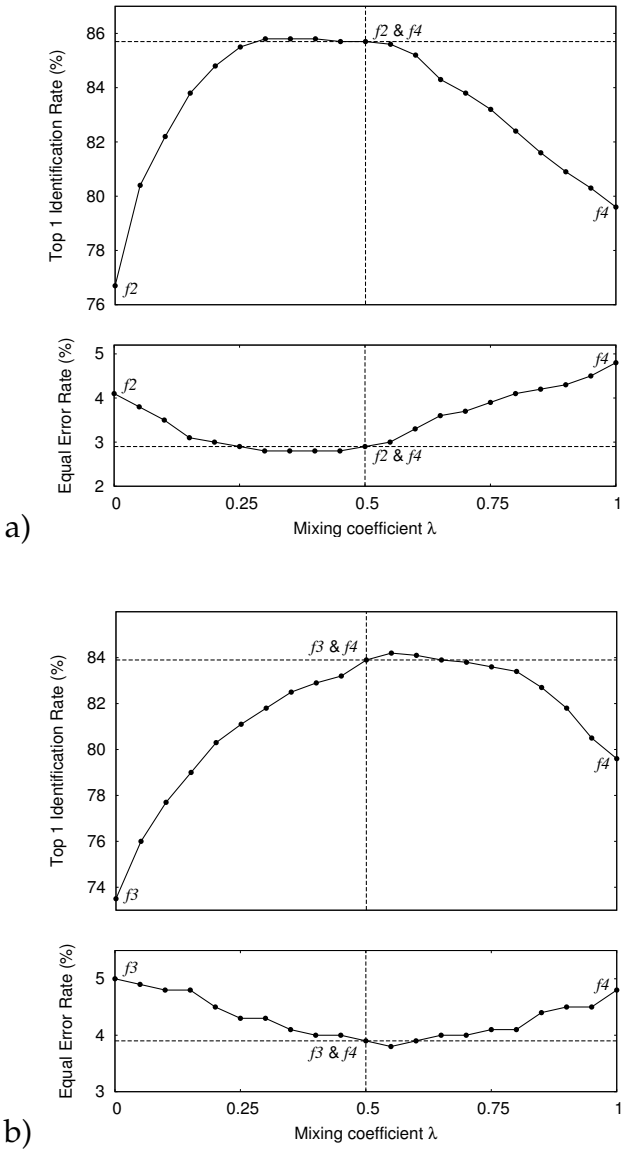

Figure 5.11: *Writer identification and verification performance on the Large dataset for a weighted combination of features a) f2 and f4, b) f3 and f4. Only marginal improvements are obtainable over the performance levels of the simple average combination represented by the horizontal lines and corresponding to a mixing coefficient* $\lambda = 0.5$.

Fig. 5.12 gives a graphical overview of the writer identification results on the Large dataset for individual features and for the best performing feature combination. The Top-1 and Top-10 recall rates were used as anchor points in reporting the numerical results from tables 5.3 and 5.4. Fig. 5.13 gives the writer verification ROC curves. In our case, the EER values are sufficiently descriptive, as a performance measure, for the whole profile of the corresponding ROC curves.

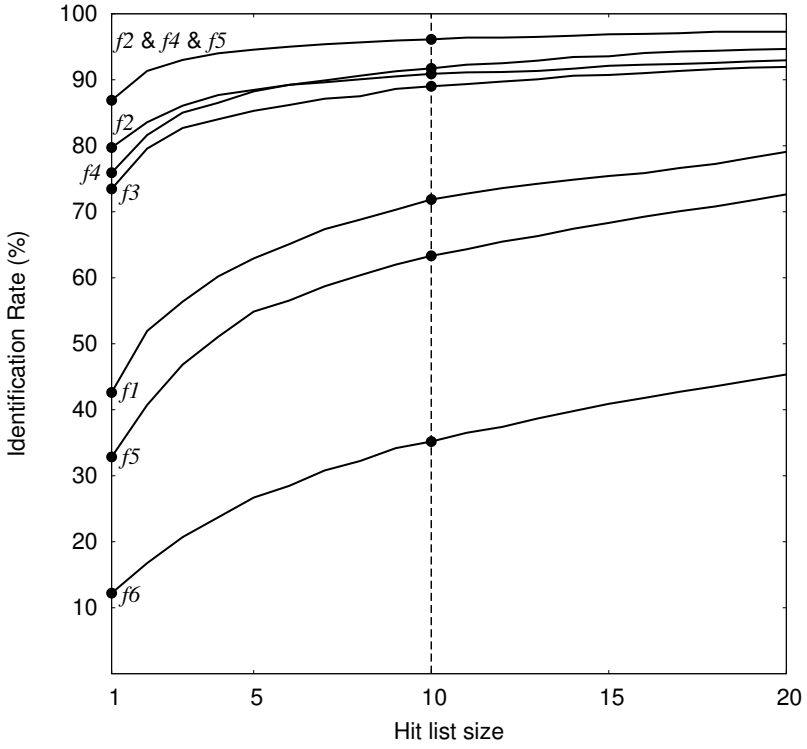

Figure 5.12: *Writer identification performance as a function of hit list size. The results were obtained on the Large dataset containing 900 writers, 2 samples per writer.*

5.7 Discussion

The analyzed features are not complete: feature extraction is a lossy operation and thus starting from the feature values, a total reconstruction of the input handwriting image is not possible. On the other hand, this is also not desirable, as we are interested in text-independent methods for writer identification and verification. Our features used to encode individual handwriting style are independent of the textual content of the handwritten sample. The handwriting is merely seen as a texture characterized by joint directional probability distributions or as a simple stochastic shape-emission process characterized by a grapheme occurrence probability.

The directional PDFs (*f1*, *f2*, *f3*) operate at the scale of the ink-trace width and implement a local phase analysis yielding results that are significantly better than those of the autocorrelation feature (*f6*) capturing amplitude information. The writer-specific shape-emission PDF (*f4*) operates at the scale of characters. Combining information across multiple scales by feature fusion results in sizeable performance improvements. The presented fusion method based on simple distance averaging diminishes the risk of a biased solution, while capturing most of the achievable increases in writer identification and verification performance.

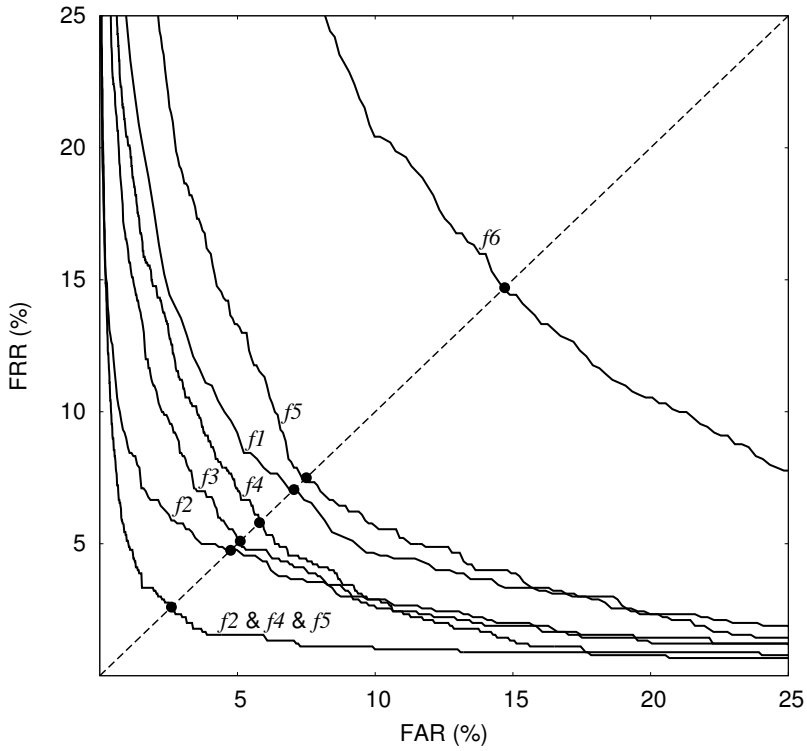

Figure 5.13: *Writer verification ROC curves obtained on the Large dataset containing 900 writers, 2 samples per writer. The EER operational points lie on the dotted diagonal.*

Similar to the previous chapter, we accomplished a more in-depth analysis of the performance of our allograph-level method on the Large dataset. The computation of feature *f4* depends on two important issues: the size of the shape codebook and the clustering algorithm used to generate the codebook. We have run large-scale computational experiments to compare three clustering methods over a large range of codebook sizes: k-means, Kohonen SOM 1D and 2D. Figures 4.1, 4.2 and 4.3 show examples of shape codebooks that have been generated by the three clustering methods. Figure 5.9 shows examples of codebooks of increasing size generated by k-means clustering.

In the experiments, the number of clusters used was varied from 9 (3x3) to 2500 (50x50). A number of 200 epochs have been used for training the Kohonen SOMs. Computations have been run on a Beowulf high-performance Linux cluster with 1.7GHz / 0.5GB nodes. Training times for codebooks of size 400: k-means - 1 hrs, ksom1D - 10 hrs, ksom2D - 17 hrs. Computation times for the grapheme emission PDF on codebooks of size 400: k-means - 0.5 s / sample, ksom1D - 1.5 s / sample, ksom2D - 3.1 s / sample. These computation times were obtained using the 'gcc' compiler with optimization for single-precision floating-point calculations.

The results obtained on the Large dataset confirm our previous findings from Chap-

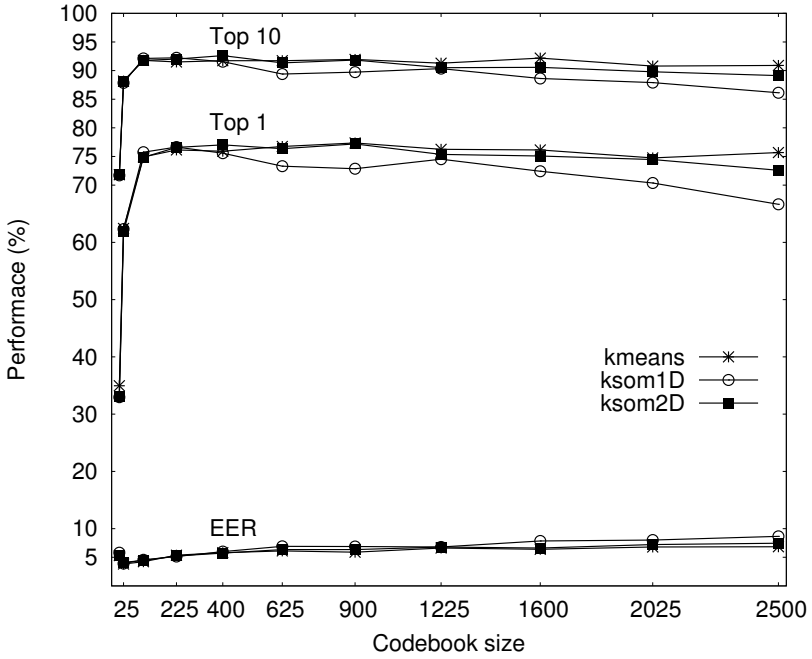

Figure 5.14: *Performance vs. clustering method and codebook size for the grapheme-based writer identification and verification method (feature f4) on the Large dataset.*

ter 4. Fig. 5.14 shows that the same performance is achieved by all three clustering methods and that performance is stable over the range of codebook sizes covered in the experiments. Writer identification rates (Top-1 and Top-10) reach a plateau for codebook sizes larger than about 100 (10x10) shapes. The writer verification EER reaches a minimum of about 4% for a codebook size of 100 and increases to about 7% for larger codebooks.

These results can be explained considering that, as the codebook size increases, it contains a larger variety of shapes and therefore becomes more discriminatory between writers, with the inevitable drawback that PDF estimation becomes more difficult given the limited amount of handwriting present in our samples. As observed also previously in the experiments reported in Chapter 4, the increase in the EER is probably due to the fact that, for larger codebooks, the dimensionality of the grapheme emission PDFs increases and consequently a unique decision threshold is no longer appropriate for all the sample-to-sample distances used in writer verification. The writer verification system commits to a global decision threshold before actually being confronted with the two samples that must be compared. An individualized threshold would be required, taking into account the within-writer variability specific to the two samples being matched in a chosen writer verification trial. However, considering the limited amount of handwritten material contained in our samples, estimating this within-writer variability is a difficult problem that requires further research. It is important to observe

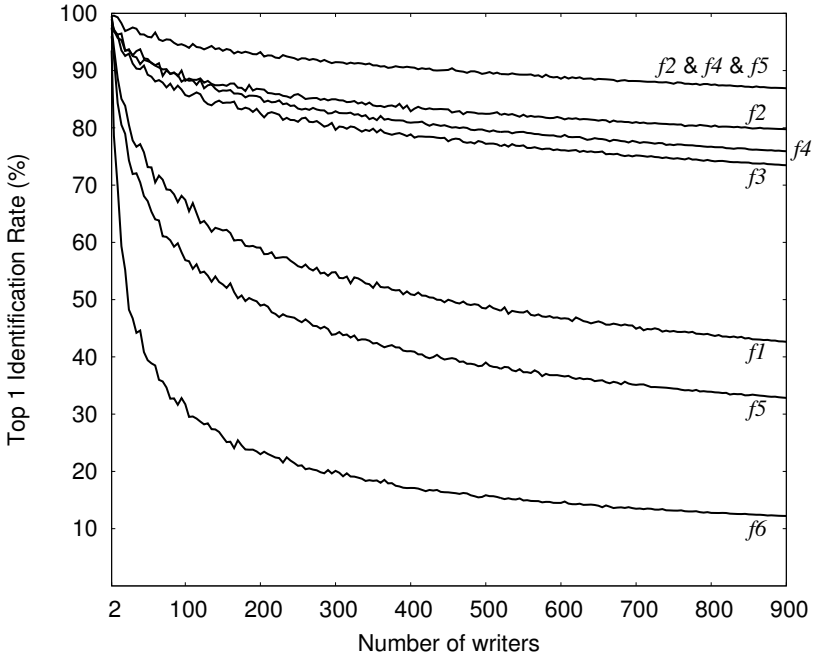

Figure 5.15: *Top-1 identification rate vs. number of writers contained in the test. For every size of the writer set, the results were averaged over fifty random draws from the Large dataset. For the complete dataset (1800 samples by 900 writers), the writer identification percentages are subject to a $\pm 3\%$ confidence interval at a 95% confidence level.*

that the described dimensionality problem does not significantly affect the writer identification performance because the query sample constitutes a vantage point with respect to which the distance rankings of the other samples remain essentially stable with the increase in codebook size. Similar results were found also on the other test datasets in the previous chapter (Bulacu and Schomaker 2005a).

The results reported for the grapheme-emission PDF (feature *f4*) in the previous sections of the chapter were obtained using a codebook generated by k-means clustering and containing 400 graphemes, which was chosen as an anchor point. The grapheme codebook is obtained much faster using k-means instead of Kohonen training.

The grapheme codebook spans up the shape space of the possible allographic parts encountered in handwritten samples as a result of the ink segmentation procedure. The three clustering methods considered here seem to perform equally well the task of selecting representative graphemes adequate for constructing a shape-occurrence PDF informative about the writer identity.

We can confidently conclude that the proposed allograph-level method is robust to the underlying shape representation used (whether contours or normalized bitmaps), to the size of codebook used (stable performance for sizes from 10^2 to 2.5×10^3) and to the clustering method used to generate the codebook (essentially the same performance

was obtained for k-means, ksom1D and ksom2D).

In order to complete our study, another necessary analysis was carried out evaluating how the identification performance (Top-1 and Top-10) depends on the number of writers contained in the test dataset. We determined this relationship by experiment using the Large dataset: for each size of the writer set (up to 900 writers), fifty identification tests were performed on random selections of writers and the results were averaged. Fig. 5.15 shows the Top-1 identification rate as a function of the number of writers for individual features and for the feature combination *f2* & *f4* & *f5*. Naturally, the identification rate decreases as the number of writers grows. However, the decline is not severe. In the range studied, for the best performing feature combination *f2* & *f4* & *f5*, we observe that the Top-1 identification rate drops by approx. 2-3% for every doubling of the number of writers in the dataset. Our writer identification system shows usable performance for 10^3 writer sets. Undoubtedly, further experiments with larger numbers of writers are needed in order to approach the 10^4 scale of the actual forensic databases.

The writer identification experiments reported in this thesis always involved two samples per writer: one was used as the query, while the other one represented the correct hit that the system was supposed to find in the database. Having more samples per writer enrolled in the database, increases the chance of finding in the top positions of the hit list the correct author for a given query. We have run writer identification tests on the original IAM database that included at least 3 samples per writer for about a quarter of the total of 650 writers incorporated in the set. For the best performing feature combination *f2* & *f4* & *f5*, we obtained writer identification rates of Top-1 92% and Top-10 98%. These values exceed the identification rates obtained on our modified IAM set that always contained only two samples per writer (see Table 5.4).

In another study performed on a subset comprising 100 writers from the Firemaker dataset, our methods largely outperformed two actual systems used in current forensic practice (Schomaker and Bulacu 2004). The use of automatic and computation-intensive approaches in this application domain will allow for massive search in large databases, with less human intervention than is current practice. By reducing the size of a target set of writers, detailed manual and microscopic forensic analysis becomes feasible. In the foreseeable future, the toolbox of the forensic expert will have been thoroughly modernized and extended. Part of our directional texture-level features have already been included in real-life applications.

It is important to note that the methods described in this thesis are equally applicable to handwriting as well as machine print: writer identification vs. font identification (e.g. for OCR). Besides the forensic field, interesting potential applications are in the domain of historic document analysis: identification of scribes or manuscript dating on medieval handwritten documents or identification of the printing house on historic

prints. Furthermore, writer identification may be used in handwriting recognition as a preprocessing step allowing the use of dedicated recognizers specialized to one writer or to a limit group of writers with similar handwriting styles.

5.8 Conclusions

The writer identification and verification methods described in this thesis exploit two essential sources of behavioral information regarding handwriting individuality. Firstly, habitual pen grip and preferred writing slant and curvature are reflected in the directional texture-level features that operate in the angular domain at the scale of the ink-trace width. Secondly, the personalized set of allographs that each person uses in writing is captured by the grapheme occurrence probability. This feature works in the Cartesian domain at the scale of the character shapes.

The proposed features are probability distributions extracted from the handwriting images and offer a text-independent and robust characterization of individual handwriting style. They have practical feasibility and they are applicable to free-style handwriting, both cursive and isolated. Combining texture-level and allograph-level features yields very high writer identification and verification performance, with usable rates for datasets containing 10^3 writers.

The challenge is to integrate the recent developments in this field of behavioral biometrics into the real writer identification systems of the future.